FACULTATEA DE AUTOMATICA SI CALCULATOARE

Referat Laboratorul 4 - Tehnologii avansate utilizate in sistemele cu procesoare grafice

Deaconu Ioan
May 3, 2015

1 INTRODUCERE

Componenta cu cea mai rapida evolutie dintr-un sitem de calcul, placa video datoareaza acest lucru industriei jocurilor video.

Placa video, ca orice alta componenta periferica din calculator, a aparut din nevoia de a avea o puere de procesoare cat mai mare. Prima placa video din lume, creata de IBM in 1981, avea o singura functionalitate, afisarea a 25 de linii a cate 80 de caractere fiecare. Desi nu pare un lucru avansat, la vremea respectiva era un progres tehnologic imens deoarece a mutat randarea imaginilor de pe procesor catre un echipament periferic dedicat. Urmatorul pas a fost randarea de imagini, lucru permis de catre iSBX 275 Video Graphics. Aceasta placa video, aparuta in 1983, suporta rezolutii de pana la 256x256 pixeli si 8 culori.

Aceasta placa video a permis aparitia jocurilor video pentru PC. Industria jocurilor a evoluat rapid, cerand placi video din ce in ce mai rapide, astfel in 1995, au aparut primele placi video capabile sa accelereze hardware randarea 3D. Desi erau capabile de randare 3D, fiecare producator avea un api 3D diferit de ceilalti producatori, iar din aceasta cauza, Microsoft a lansat un api comun in anul 1995, numit DirectX.

2 NOTIUNI TEORETICE AFERENTE

Unitatile functionale ale unei placi video sunt:

- Pixel Shader.
- Vertex Shader.

- Geometry Shader.
- Tessellation Shader.

2.1 Pixel Shader

Pixel shader este o unitate care se ocupa de culoarea pe care o are un pixel al ecranului. Acesta unitate aplica efecte peste culoare de baza, efecte precum blur, shadowing, edge detection. Din acest motiv, un pixel shader trebuie sa stie contextul pixelului pentru a putea calcula culoarea finala.

2.2 Vertex Shader

Vertex shader este o unitate care poate calcula pozitia unui vertex. Mai exact, scopul lui este sa transforme coordonatele 3D ale unui vertex in spatiul 2D corespunzator al ecranului. Pentru a realiza acest lucru, un shader vertex poate lucra cu proprietatii ale liniei cum ar fi culoarea, pozitia si coordonatele texturii, dar aceasta unitate nu poate crea vertexi noi. Rezultatul unui vertex shader poate fi trimis unui Geometry Shader sau unui engine de Rasterizare.

2.3 Geometry Shader

Geometry Shader, adaugat in DirectX10, spre deosebire de Vertex Shader, poate crea vertexi noi. Acesta primeste ca intrare un poligon, si poate sa adauge puncte noi pentru a crea poligoane noi si un nivel mai mare de detaliu. Rezultatul acestei unitati este trimis lui Pixel Shader [5].

2.4 Tessellation Shader

Tessellation Shader, introdus in DirectX11, permite obiectelor aflate in apropierea camerei sa fie alcatuite din mai multe poligoane fata de obiectele mai indepartate. Acest lucru este realizat hardware, si doar de catre acesta unitate, astfel, impactul asupra performantei este mic comparativ cu nivelul de detaliu generat.

3 Implementare

Placile video se ocupa in marea partea a timpului cu randarea imaginii care se afiseaza pe ecran. Acest lucru presupune generarea unei imagini care contine foarte multi pixeli la un refresh rate de aproximativ 60 de cadre pe secunda.

Doar pentru a genera 60 de cadre pe secunda, este nevoie de un bus capabil sa trimita date cu o viteza de minim 4GBit/s.

Fiecare cadru este compus din aproximativ 2 milioane de pixeli. Aceasta viteza este atinsa datorita faptului ca fiecare pixel este calculat individual, placile video fiind un exemplu ideal de procesare paralela. Incepand cu api-ul DirectX10, sau arhitectura G80 , placile video au incorporat Vertex Shader si Pixel Shader intr-o singura unitate numita Cuda Core [6]. Astfel toate operatiile de tip Vertex Shader sau Pixel shader sunt realizate de aceasi unitate Cuda

Core. Acest lucru elimita cazul in care daca se executau operatii de tip vertex, doar Vertex Shaderul sa fie operational si Pixel Shaderul sa fie idle, si invers.

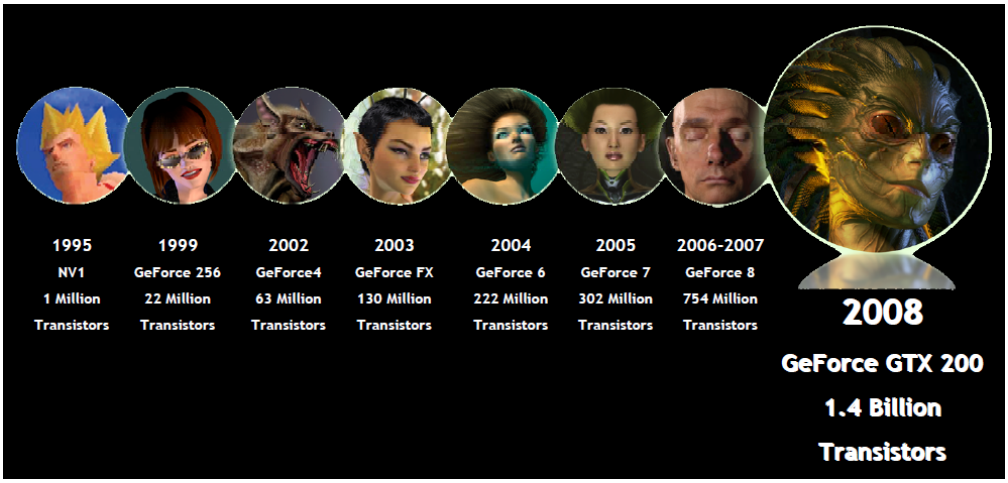

Figure 3.1: Evolutia placilor video 3D

3.1 CUDA CORE

Spre deosebire de un procesor clasic, care se bazeaza pe un numar mic de nuclee, intre 1 si 16, un procesor grafic foloseste un numar mult mare, de exemplu 3072 de Cuda Cores, in cazul placi Nvidia Geforce Titan X [4], dar si de o latime de banda net superioara unui procesor clasic, o latime de 336.5 GB/s.

Fiecare Cuda Core se aseamana ca functionalitate in mare parte cu un nucleu al unui procesor normal, dar acesta ruleaza la o frecventa mult mai mica, de aproximativ 3-4 ori mai mica, si au functionalitati limitate, pentru a mentine suprafata fizica a acestuia cat mai mica.

Deoarece numarul nucleelor Cuda este atat de mare, multihreading-ul este implementat hardware, pentru a reduce cat mai mult overhead-ul datorat de planificarea threadurilor. Pentru a simplica aceasta planificare, nucleele sunt grupate in clustere, fiecare cluster executand aceeasi intructiunie. Acest lucru este posibil deoarece procesoarele video se bazeaza pe o arhitectura de tip SIMT - Single Instruction Multiple Threads [6].

3.2 ARHITECTURA FERMI - 2010

Arhitectura Fermi a continuat arhitectura G80, astfel ca fiecare Cluster de Cuda Core, sau Procesoarea Stream, contin 32 de procesoare Cuda. Acestea au ramas la fel de complexe ca cele anterioare dar numarul lor a crescut la 512.

3.3 ARHIRECTURA KEPLER - 2012

In arhitectura Kepler, numarul de Procesoare Stream a scazut de la 16 la 8, astfel fiecare cluster continea 192 de procesoare Cuda. Pe langa numarul crescut de procesoare Cuda, acesteau au

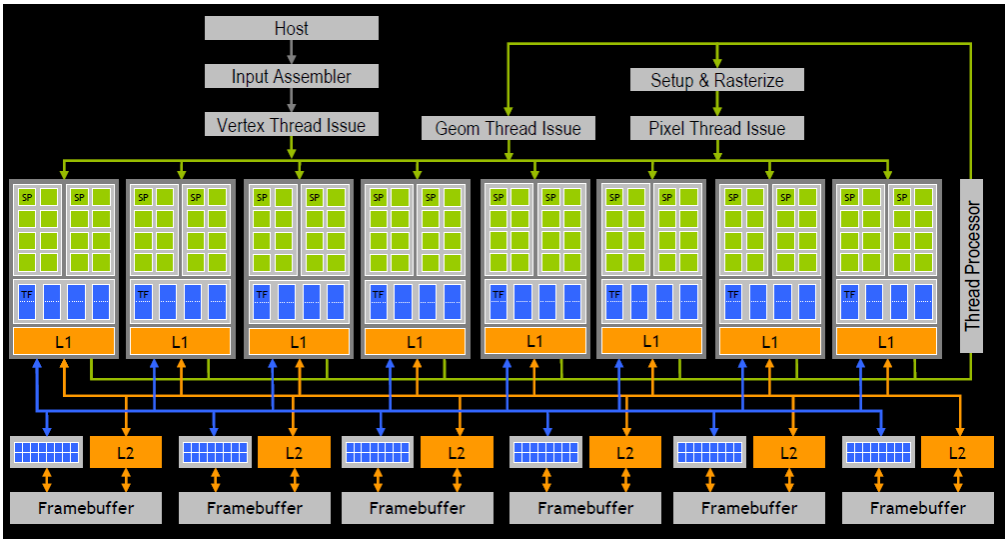

Figure 3.2: Arhitectura Nvidia G80

fost simplicate, pentru a obtine o viteza de executie mai mare dar si un consum mai redus. In arhitectura Fermi, procesoarele Cuda functionau la o frecventa dubla fata de cea a GPU-ului, dar in acesta arhitectura, deoarece functioneaza la o frecventa mult mai redusa, consumul lor este cu 90% mai mic decat cele ale arhitecturii precedente. Acest lucru a permis triplarea procesoarelor Cuda de la 512 la 1536.

3.4 Arhitectura Maxwell - 2014

Ducand mai departe ideea de performanta marita pentru un consum redus, arhitectura Maxwell a eficientizat arhitectura Kepler, astfel ca 128 de Cuda Cores Maxwell sunt la fel de rapide ca 192 de Cuda Cores de tip Kepler. Deasemenea a implementat un algoritm de compresie a texturilor, ceea ce a permis pastrarea aceleiasi latimi de banda a memoriei cu o creste a performantei.

4 Simulare

Datorita faptului ca o placa video trebuie sa execute miliarde de calcule matematice, ray tracing, etc, acesa arhitectura este optimizata pentru acest timp de calcule. Datorita acestui lucru, toate placile video care sunt compatibile cu tehnologii de tip Cuda sau OpenCL, tehnologii ce pot acelera calcule matematice, in special lucrul cu matrici. Viteza cu care aceste operatii sunt executate este motivul pentru care toate super calculatoarele construite in ultimi 5 ani au, pe langa mii de procesoare clasice, si placi video profesionale. Diferenta intre o placa video profesionala si una clasica este ca aceasta poate sa execute si operatii matematice in virgula mobila - Figura 4.1. Mai bine zis, pe langa operatii matematice cu operatori de tip float, aceasta poate accelera foarte bine si operatii matematice cu operatori de tip double.

Dupa cum se poate observa in testul de mai jos, Fig 4.1, performanta teoretica a placilor video se dubleaza odata la fiecare an, fata de procesoarele clasice care dubleaza peformanta odata la circa 2-3 ani de zile. Deasemenea, pe langa performanta imbunatatita, eficienta energetica este si ea din ce in ce mai buna, iar datele din Figura 4.2, obtinuta dintr-o prezentare Nvidia, demonstreaza acest lucru .

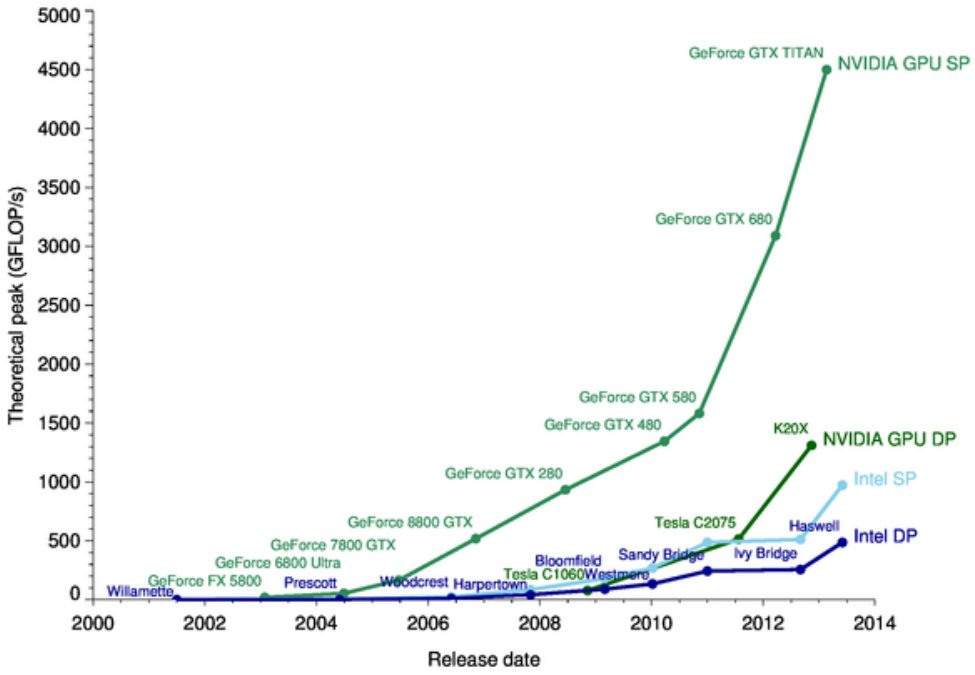

Figure 4.1: Performanta placilor grafice comparativ cu cea a procesoarelor

Putem compara si putera placilor video, care, a tot scazut in conditiile in care perfomanta a crescut. Astfel, Nvidia GTX 580, placa cu arhitectura Fermi, are un consum de 244 de Watti[1], Nvidia GTX 680, are un consum de 194 Watti[2], iar Nvidia GTX 980 are un consum de 165 Watti, in conditiile in care numarul de Nuclee Cuda a crescut de la 512 la 1536, ajung la GTX 980 sa fie in numar de 2048 [3].

5 Concluzii

Desi intial au fost gandite ca accesorii pentru pasionatii de jocuri, odata cu trecerea timpului si cresterea nevoii de a avea o putere de calcul cat mai mare, placile video au evoluat de la simple adaptoare grafice, care putea sa afiseze o imagine pe ecran, pana la acceleratoare de calcul. Aceste noi tehnologii prezente in GPU-urile actuale permit accelerarea calculelor, astfel ca programe de genul Matlab, programe de randare grafica, unde trebuie calculata interactiunea fiecarei raze de lumina cu fiecare obiect, si chiar simulari de fizica a obiectelor sa fie calculate in timp de real. Consumul lor este in continua scadere, ceea ce a permis placilor grafice mobile

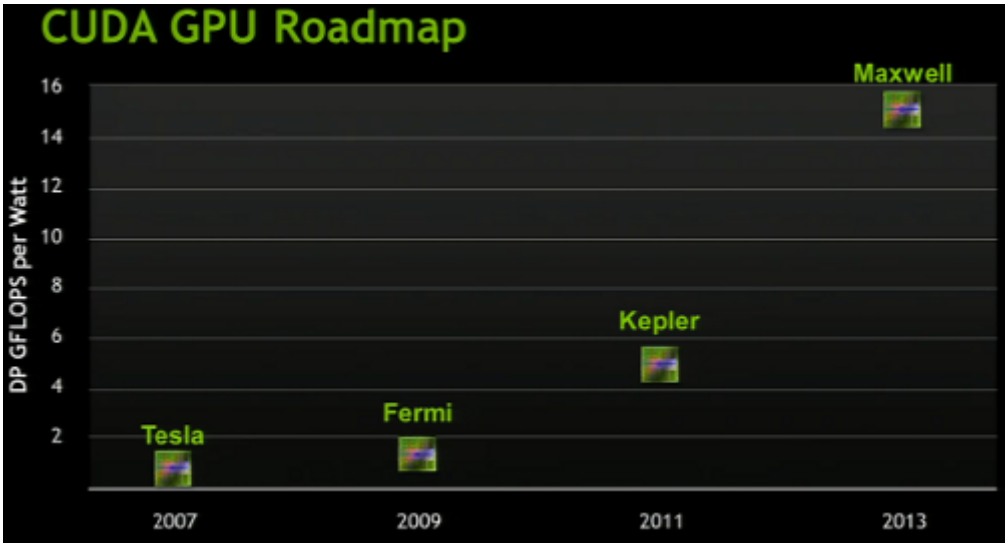

Figure 4.2: Eficienta arhitecturiilor Fermi, Kepler si Maxwell

sa fie si ele din ce in ce mai rapide, scopul celor de la Nvidia fiind ca intr-un viitor apropiat, sa nu mai existe nici o diferenta de performanta intre placile mobile si cele desktop.

Desi sunt de sute de ori mai rapide ca un procesor in aceste programe, placile video inca nu sunt suficient de avansate incat sa poata rula independent, ele find specializate pe calcule matematice si grafica video.

REFERENCES

[1] Nvidia geforce gtx 580 - specifications. `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-580/specifications`. Accessed: 2015-05-3.

[2] Nvidia geforce gtx 680 - specifications. `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-680/specifications`. Accessed: 2015-05-3.

[3] Nvidia geforce gtx 980 - specifications. `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-980/specifications`. Accessed: 2015-05-3.

[4] Nvidia geforce titan x - specifications. `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specificationsi`. Accessed: 2015-05-3.

[5] D. Kirk et al. Nvidia cuda software and gpu parallel computing architecture. In *ISMM*, volume 7, pages 103–104, 2007.

[6] D. Luebke. Gpu architecture: Implications & trends. *ser. SIGGRAPH*, 2008, 2008.